# Secure Login and Role-Based Access in ASP.NET Core MVC

Name: Souvik Pradhan

Preptest - Wipro NGA .Net Fullstack Angular

Date: August 28, 2025

# Introduction

This project demonstrates how to implement authentication and role-based access control in an ASP.NET Core MVC application. The application provides secure login, role assignment for Admin and User roles, and protection against common vulnerabilities such as CSRF and unencrypted data transmission.

# Implementation Steps

Step 1: Create a new ASP.NET Core MVC project using Visual Studio or CLI.

Step 2: Configure Identity with UserManager, SignInManager, and RoleManager.

Step 3: Implement secure login page and role-based access logic in AccountController.

Step 4: Create views: Login, AdminDashboard, and UserProfile.

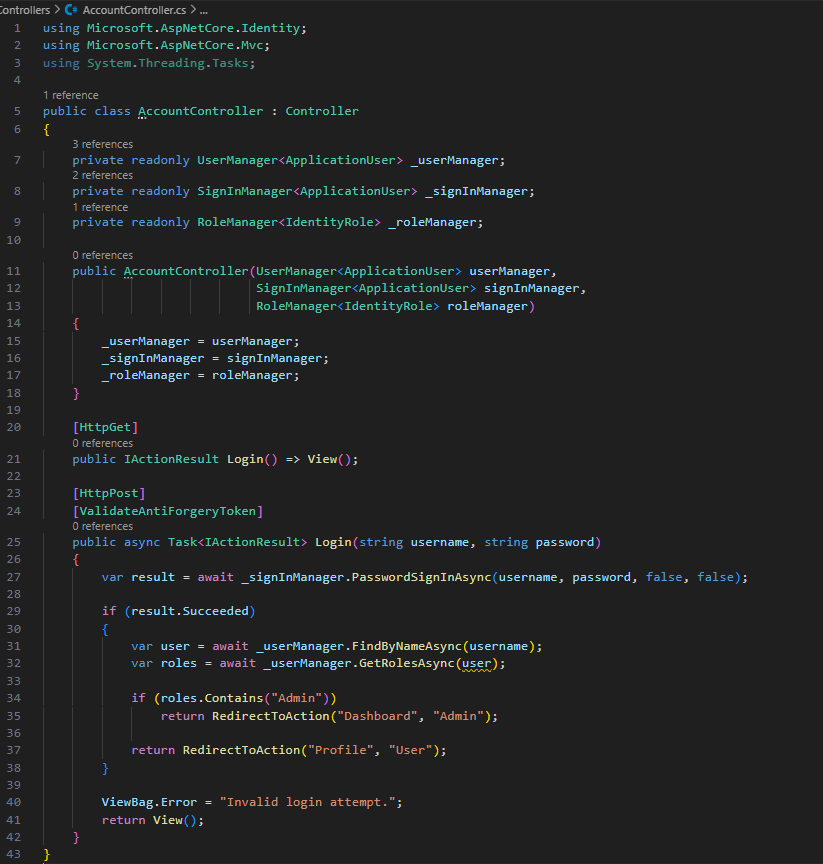
Step 5: Use [Authorize] attributes to restrict access to views based on roles.

Step 6: Enable HTTPS and AntiForgeryToken for additional security.

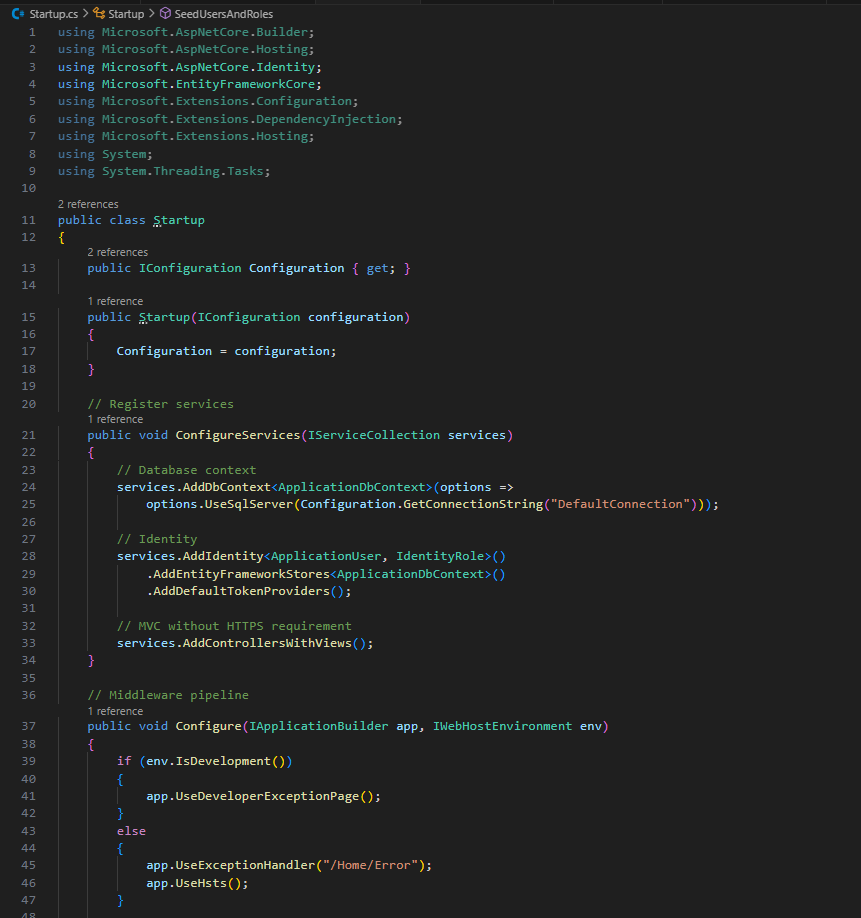
# Code Implementation

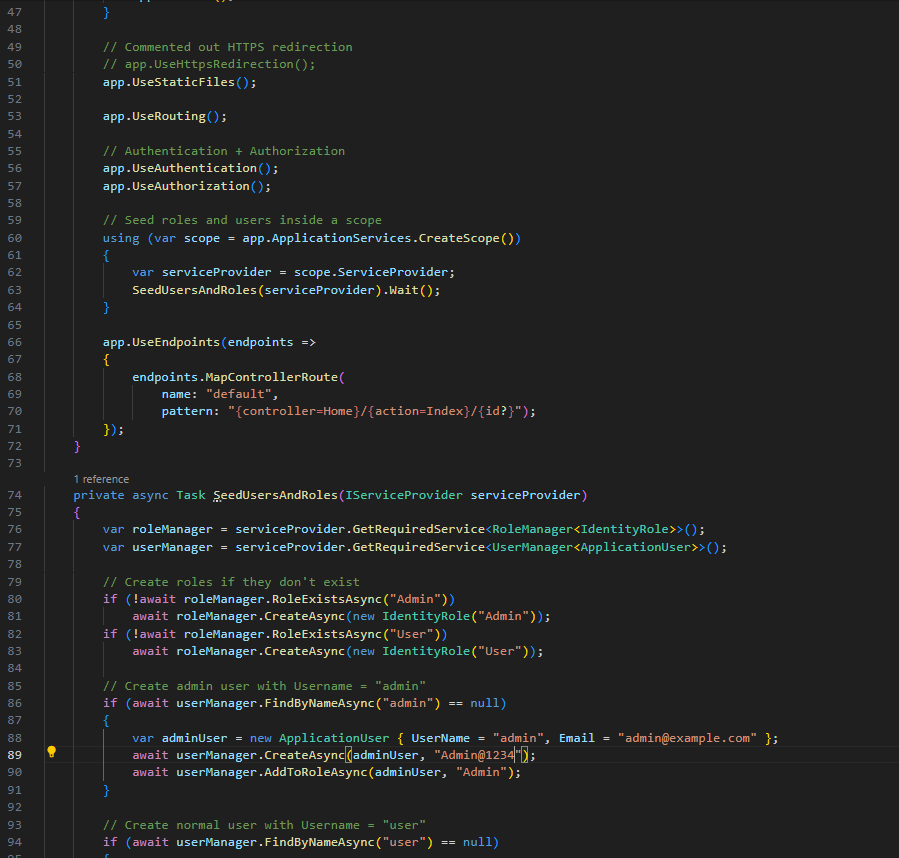
Below are key code snippets from the project:

**AccountController.cs-**

****

**Startup.cs-**

****



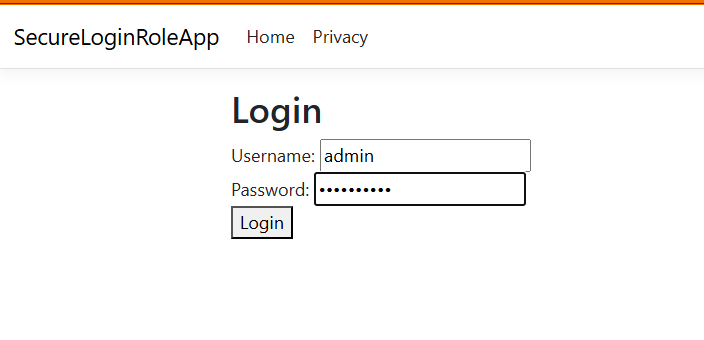
**Login.cshtml-**



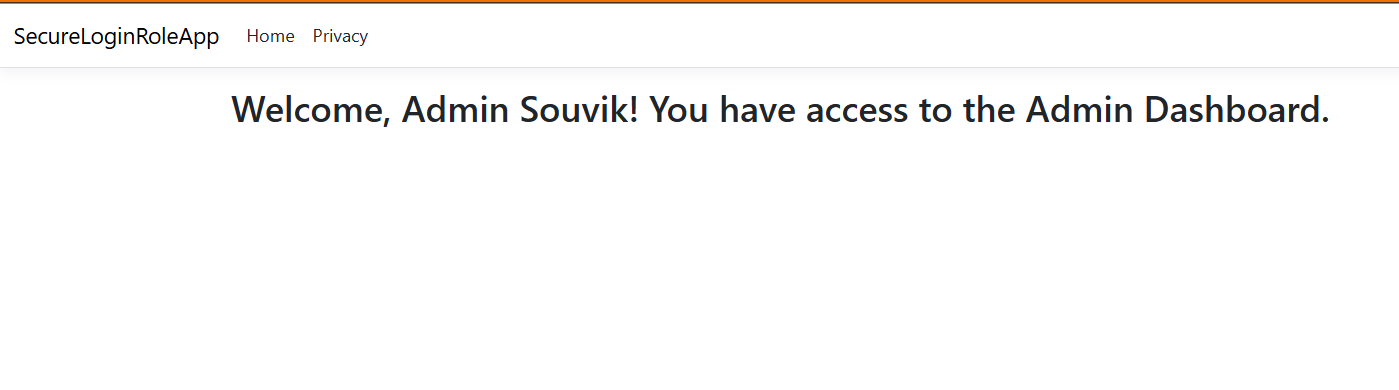
# Execution and Output

The following screenshots demonstrate the working of the project:

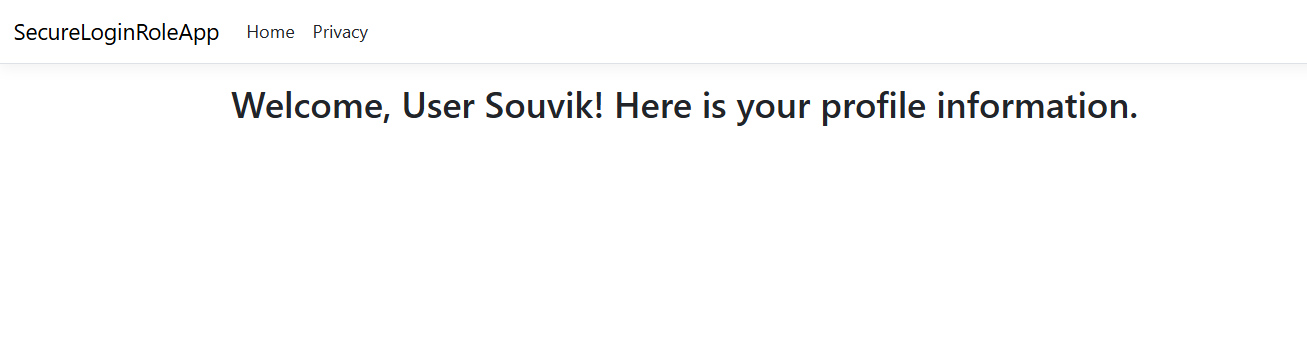
**Login page-**

****

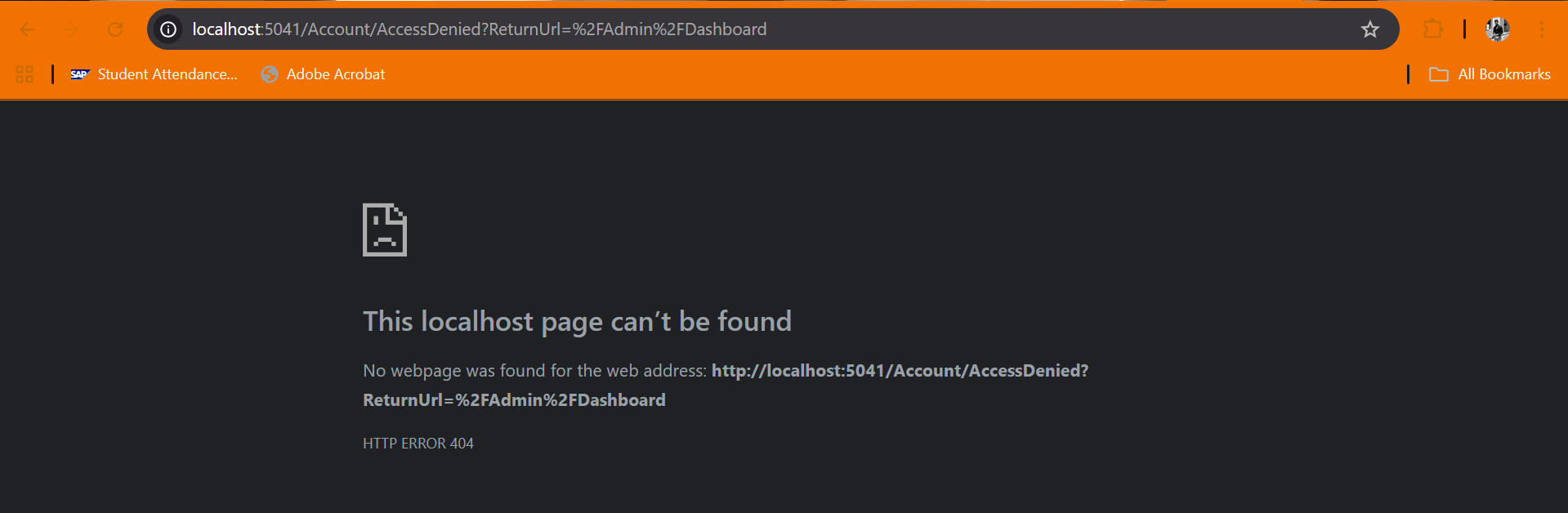
**Successful Admin login -AdminDashboard-:**

****

**Successful User login – UserProfile-:**



**Access Denied message when User tries to access AdminDashboard-:**



# Conclusion

This project successfully demonstrates how authentication and role-based authorization can be implemented in ASP.NET Core MVC applications. With secure login, role separation, and protection mechanisms like HTTPS and CSRF tokens, the application ensures a strong security model.